IN THE CLAIMS

Please amend the claims to the following.

1. (Currently Amended) A method comprising:

   receiving a system management interrupt (SMI);

   checking a ~~the~~ state of a second processor with a first processor; and

   ~~(a) if~~ in response to the state of the second processor being ~~is~~ inactive, handling the SMI with the first processor;

   ~~(b) if~~ in response to the state of the second processor ~~is~~ being active and not in SMI mode (SMM), waiting for the second processor to enter SMM ~~SMI mode~~;

   ~~(c) if~~ in response to the state of the second processor being ~~is~~ active and in SMM ~~SMI mode~~, handling the SMI on both the first and the second processors.

2. (Original) The method of claim 1, wherein checking the state of a second processor with a first processor comprises: examining a storage medium with the first processor, wherein the storage medium stores values representative of the second processor's state.

3. (Original) The method of claim 2, wherein the storage medium is system memory.

4. (Original) The method of claim 3, wherein the system memory stores values representative of the second processor's state in a synchbyte.

5. (Original) The method of claim 4, wherein the synchbyte, when having a first value, represents that the second processor is in an inactive state.

6. (Currently Amended) The method of claim 4, wherein the synchbyte, when having a second value, represents that the second processor is active but not in SMM ~~SMI mode~~.

7. (Currently Amended) The method of claim 4, wherein the synchbyte, when having a third value, represents that the second processor is in an active state and in SMM ~~SMI mode~~.

8. (Original) The method of claim 2, wherein the storage medium is a register.

9. (Currently Amended) The method of claim 8 ~~2~~, wherein the register is located in the second processor.

10. (Original) The method of claim 2, wherein the storage medium's default value represents an inactive state for the second processor.

11. (Original) The method of claim 2, further comprising: updating the storage medium with the second processor to reflect the second processor's current state.

12. (Currently Amended) The method of claim 11, wherein updating the storage medium comprises: writing a value to the storage medium to represent an inactive state, ~~if~~ in response to the second processor ~~is~~ going into a low power state.

13. (Currently Amended) The method of claim 11, wherein updating the storage medium comprises: writing a value to the storage medium to represent an active and not in SMM ~~SMI mode~~ state, ~~if~~ in response to the second processor ~~is~~ waking-up and not being in SMM ~~SMI mode~~.

14. (Currently Amended) The method of claim 11, wherein updating the storage medium comprises: writing a value to the storage medium to represent an active and in SMM ~~SMI mode~~ state, ~~if~~ in response to the second processor ~~is~~ entering SMM ~~SMI mode~~.

15. (Original) The method of claim 1, further comprising: generating the SMI before receiving the SMI.

16. (Original) The method of claim 15, wherein generating the SMI is done through software.

17. (Original) The method of claim 15, wherein generating the SMI is done through hardware.

18. (Original) The method of claim 1, wherein the first and second processors are logical processors.

19. (Original) The method of claim 1, wherein the first and second processors are physical processors.

20. (Withdrawn) A method comprising:

assigning a first memory space for system management to a first processor; and

assigning a second memory space for system management to a second processor, wherein the second memory space partially overlaps the first memory space leaving at least a first non-overlapping region.

21. (Withdrawn) The method of claim 20, wherein the overlap of the first and second memory space also leaves a second non-overlapping region.

22. (Withdrawn) The method of claim 20, wherein the size of the first non-overlapping region is at least the size of a save-state area for the first processor.

23. (Withdrawn) The method of claim 21, wherein the size of the second non-overlapping region is at least the size of a save-state area for the second processor.

24. (Withdrawn) The method of claim 23, wherein the size of first and second non-overlapping regions are at least the size of a system management interrupt (SMI) handler code for the second processor.

25. (Withdrawn) The method of claim 20, wherein the overlapping region contains a synchronization area to store the system management state of at least the second processor.

26. (Withdrawn) The method of claim 20, wherein the overlapping region contains the save-state area for the first processor.

27. (Withdrawn) The method of claim 26, wherein the first non-overlapping region contains the save-state area for the second processor.

28. (Withdrawn) The method of claim 20, wherein the first and second memory spaces are in system memory.

29. (Withdrawn) The method of claim 20, wherein the first and second processors are logical processors.

30. (Withdrawn) The method of claim 20, wherein the first and second processors are physical processors.

31. (Withdrawn) A microprocessor comprising:

a first logical processor and a second logical processor coupled to a storage medium, wherein the storage medium stores the system management state of the second logical processor.

32. (Withdrawn) The microprocessor of claim 31, wherein the storage medium is a register in one of the processors.

33. (Withdrawn) The microprocessor of claim 31, wherein the storage medium represents that the second processor is in an inactive state.

34. (Withdrawn) The microprocessor of claim 31, wherein the storage medium represents that the second processor is in an active and not in SMI state.

35. (Withdrawn) The microprocessor of claim 31, wherein the storage medium represents that the second processor is in an SMI mode.

36. (Withdrawn) An apparatus comprising:

a storage medium, coupled to a first and second logical processor, having a first memory range assigned to the first processor for system management and a second memory range assigned to the second processor for system management, wherein the first and second memory ranges partially overlap leaving a first and second non-overlapping range.

37. (Withdrawn) The apparatus of claim 36, wherein the size of the first and second non-overlapping ranges are at least the size of the first and second processor's save-state area respectively.

38. (Withdrawn) The apparatus of claim 36, wherein the first and second processors are logical processors.

39. (Withdrawn) The apparatus of claim 36, wherein the first and second processors are physical processors.

40. (Withdrawn) The apparatus of claim 36, wherein the overlapping region has a synchronization area that may be modified by both the first and the second processor.

41. (Withdrawn) The apparatus of claim 40, wherein the synchronization area contains a synchbyte to represent the system management states of at least one processor.

42. (Withdrawn) The apparatus of claim 41, wherein the synchbyte is used to synchronize the first and second processors before handling a system management interrupt.

43. (Currently Amended) A system comprising:

~~a controller hub, coupled to~~ a first and a second processor; and

a storage medium, coupled to the first and the second processor, to hold ~~store the~~ a system management state of at least the second processor, wherein the first processor and the second processor, ~~checks the system management state of the second processor after~~ in response to receiving a ~~first~~ system management interrupt (SMI), ~~is received~~ are to synchronize entrance into a system management mode (SMM) to handle the SMI based on the system management state of at least the second processor held in the storage medium.

44. (Currently Amended) The system of claim 43, wherein the first and the second processor to synchronize entrance into the SMM comprises: the first processor ~~handles~~ handling the SMI without waiting for the second processor to enter SMM, ~~if~~ in response to the system management state of at least the second processor indicating the second processor is inactive.

45. (Currently Amended) The system of claim 43, wherein the first and the second processor to synchronize entrance into the SMM comprises: the first processor waiting ~~waits~~ for the second processor to enter SMM ~~SMI mode~~ and update the system management state of at least the second processor to indicate the second processor has entered SMM ~~storage device, if~~ in response to the system management state of at least the second processor ~~is~~ indicates the second processor active and not in SMM ~~SMI mode~~.

46. (Currently Amended) The system of claim 43, wherein the first and the second processor to synchronize entrance into the SMM comprises: the SMI ~~is~~ being handled on ~~both~~ the first and second processors~~, if~~ in response to the system management state of at least the second processor indicating the second processor is active and in SMM ~~SMI mode~~.

47. (Currently Amended) The system of claim 43, wherein the storage medium includes a ~~is~~ system memory.

48. (Currently Amended) The system of claim 47, wherein the system memory ~~contains~~ is to hold a synchbyte that represents the system management state of ~~stores values representative of~~ at least the second processor, and wherein the storage medium further includes a cache memory coupled between the system memory and the first and the second processor to cache the synchbyte ~~processors system management state~~.

49. (Currently Amended) The system of claim 43, wherein the storage ~~device~~ medium includes ~~is~~ a register.

50. (Currently Amended) The system of claim 43, wherein the storage ~~device~~ medium includes ~~is~~ a flash memory.

51. (Currently Amended) The system of claim 48 [[43]], wherein the first and second processors are logical processors, which share access to the cache, the first processor, the second processor, and the cache located on a single package.

52. (Original) The system of claim 43, wherein the first and second processors are physical processors.

53. (Withdrawn) A system comprising:

first processor, coupled to a memory, having a first address range in the memory assigned for system management; and

a second processor, coupled to the memory, having a second address range in the memory assigned for system management, wherein the first and second address ranges partially overlap leaving a first non-overlapping range and a second non-overlapping range.

54. (Withdrawn) The system of claim 53, wherein the first and second non-overlapping ranges are at least the size of a save state range for each processor.

55. (Withdrawn) The system of claim 53, wherein the first and second non-overlapping ranges are at least the size of the system management interrupt (SMI) handler code for the second processor.

56. (Withdrawn) The system of claim 53, wherein the memory is system memory.

57. (Withdrawn) The system of claim 53, wherein the first and second processors are logical processors.

58. (Withdrawn) The system of claim 53, wherein the first and second processors are physical processors.